ARMY RESEARCH LABORATORY

Graphical User Interface for ZEUS

Harris Edge
Jerry Clarke

ARL-TR-1093 June 1996

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

NOTICES

ERRATA RE:

ARL-TR-1093, "Graphical User interface for ZEUS," by
Harris Edge and Jerry Clarke, dated June 1996

Request recipient make the following pen and ink changes to subject report:

Page v - Figure 6. The word "form" should be changed to read "from."

Page 9 - Figure 6. The word "form" should be changed to read "from."

Page 31 - Fifth line from bottom, last sentence, "See Figure 5 for reference," should be changed to read, "See Figure 15 for reference."

Pages 31 and 32 - The last line of page 31 and the first 18 lines of page 32. All references to "Figure 14" on these lines should be changed to read "Figure 15."

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1996	Final, April 1995 - December 1995

4. TITLE AND SUBTITLE

Graphical User Interface for ZEUS

5. FUNDING NUMBERS

PR: 1L162618AH80

6. AUTHOR(S)

Harris L. Edge and Jerry Clarke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-WT-PB
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-1093

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document describes the implementation of a graphical user interface (GUI) for an existing computational fluid dynamics (CFD) code. Development of the GUI at the U.S. Army Research Laboratory (ARL) was undertaken at the request of the U.S. Army Missile Research Development and Engineering Center (MRDEC) and is funded by ARL as part of a Technology Program Annex (TPA) agreement. As a starting point for this effort, the zonal Euler solver (or ZEUS) CFD code was chosen by MRDEC to be incorporated into the GUI environment. This document serves as an introduction to the GUI and explains some of the design philosophy. The document also serves as a brief user's guide; however, it is not meant to replace the original user's guides/reports written by the original authors of the ZEUS code.

14. SUBJECT TERMS

graphical user interface, computational fluid dynamics, aerodynamics, missiles, projectiles, fins

15. NUMBER OF PAGES

34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFED	SAR

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

		Page
	LIST OF FIGURES	v
1.	INTRODUCTION	1
2.	ABOUT ZEUS	1
3.	THE GRAPHICAL USER INTERFACE	3
4.	VISUALIZATION	8
5.	APPLYING ZEUS	8
5.1	Coordinate System	10
5.2	Defining Zones	10
5.3	Boundary Conditions	13
5.4	Geometry	15
6.	SUMMARY	21
7.	REFERENCES	23
	APPENDIX: INPUT FILE VARIABLES FOR ZEUS	25
	DISTRIBUTION LIST	35

INTENTIONALLY LEFT BLANK.

LIST OF FIGURES

Figure		Page
1.	ZEUS solution flowchart	3
2a.	ZEUS graphical user interface	5
2b.	ZEUS graphical user interface	5
3.	Graphical user interface diagram	6
4.	ZEUS graphical user interface flowchart	7
5.	Pseudo-ZEUS grid	9
6.	Surface pressure contours form ZEUS solution	9
7.	ZEUS general coordinate system	11
8.	Local fin coordinate system	11
9.	General coordinate system and local coordinate system	12
10.	Zone definition	12
11.	User-defined zone spacing	14
12.	Body rotation angle	14
13.	ZEUS boundary conditions	15
14.	Body modeling	16
15.	ZEUS fin input	17
16a.	ZEUS fin example	18
16b.	ZEUS fin example	18
16c.	ZEUS fin example	19
16d.	ZEUS fin example	19
17.	Fin gap modeling	20

INTENTIONALLY LEFT BLANK.

1. INTRODUCTION

The thrust of the work described here is to further develop an existing computational fluid dynamics (CFD) code and make it more accessible to the practicing missile designer. This is accomplished by building a graphical user interface (GUI) that aids the user by providing an easy-to-understand interface with the original CFD code. The purpose of the GUI is to assist the user in all phases of code operation, including input and grid generation, code execution, and post-processing of solution data.

Development of the GUI at the U.S. Army Research Laboratory (ARL) was undertaken at the request of the Missile Research Development and Engineering Center (MRDEC) and is funded by ARL as part of a Technology Program Annex (TPA) agreement. As a starting point for this effort, the zonal Euler solver (or ZEUS) (Wardlaw and Davis 1986; Wardlaw and Priolo 1986) code was chosen by MRDEC to be incorporated into the GUI environment. ZEUS is an inviscid CFD solver. It was developed by the Navy over the past decade and is widely used within the international missile design community. To allow the engineer to interpret his or her results quickly, the GUI was designed to interface with a flow visualization program. FAST (Walatka et al. 1991), which stands for Flow Analysis Simulation Toolkit, was chosen to perform the flow visualization duties. The work on the GUI began in FY95 and resulted in the transfer of the first version of the GUI to MRDEC in early FY96. Work is continuing with further development and improvement of the GUI. It is hoped that the combination of a well-designed GUI, proven CFD solver, and a comprehensive visualization program working coherently in a single software package will be beneficial to the projectile and missile design community.

This document serves as an introduction to the GUI and explains some of the design philosophy. The document also serves as a brief user's guide; however, it is not meant to replace the original user's guides/reports written by the original authors of the ZEUS code.

2. THE ZEUS CODE

ZEUS is a zonal Euler CFD solver, which employs a second-order Godunov scheme to integrate the Euler equations and march the solution longitudinally along the body. Additional details about the integration scheme can be obtained by referencing the work of Wardlaw and Davis (1986) and Wardlaw and Priolo (1986). In terms of the application of ZEUS, there are some restrictions of its use. First, ZEUS can only be applied to cases in which the flow field is supersonic everywhere. The computational

mesh should be free of blunt discontinuities. ZEUS was written to support a zonal grid topology. The zonal topology allows fins or wings to be modeled. However, the leading and trailing edges should be fairly sharp.

The integration scheme in the ZEUS code marches the solution over the body, starting near the nosetip of the body and proceeding toward the tail. An initial solution is required to begin the computation. If the nosetip of the vehicle is sharp, a conical starting procedure can be used to generate the starting solution. In the vernacular of the GUI, it is the CONSTRT program that computes the initial solution plane for a cone and writes it to a file. If the nose is blunt, other codes must be used to generate the initial solution plane. Such codes currently exist and have been used to generate starting solutions for different nosetip geometries. Starting from the initial solution, ZEUS is executed to march the solution longitudinally along the rest of the geometry.

ZEUS can be started or stopped anywhere along the solution. In marching the length of the body, it is likely that the computational mesh will need to be refined or modified somewhere along the way. It is the user's responsibility to determine the criteria used to decide whether the quality of the grid is sufficient to provide an accurate solution. Grid modification is accomplished through the use of an auxiliary program called CONVERT, which interpolates a modified grid and solution based on grid refinement inputs. If grid refinement is necessary, the solution file written by the ZEUS code along with the new grid parameters will be used as input for the CONVERT program. The CONVERT program then produces a new solution file with the modified grid plane, called REZONE, which can be used as the initial solution plane in subsequent solution marches by the ZEUS code.

A typical application of ZEUS and its auxiliary codes, CONSTRT and CONVERT, has been placed in the form of a simple flowchart in Figure 1. In the flowchart, the CONSTRT program produces a solution file, START, which the ZEUS code uses as input. Upon execution, ZEUS produces a solution file called RESTART, which the CONVERT program uses as input. The file named START will always be used as the initial solution plane for ZEUS if the restart option is specified. If the restart option is not specified, ZEUS will use free-stream conditions for the initial solution plane, and the solution file, START, will not be needed. It is important to know the chain of events in computing a complete solution using ZEUS. The names of the files and codes listed above are represented in the GUI. However, the user must know how each code and file relate to one another.

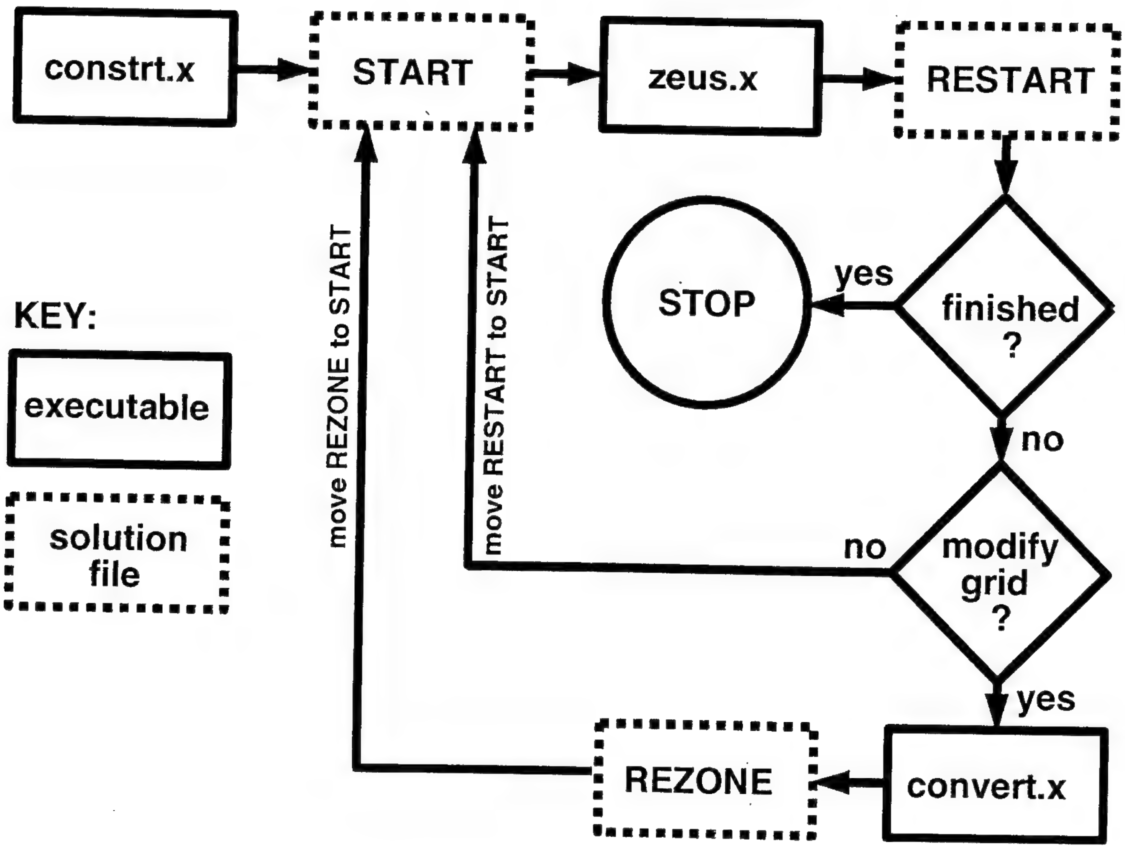

Figure 1. ZEUS solution flowchart.

3. THE GRAPHICAL USER INTERFACE

The GUI is custom-designed for the ZEUS code. It is written in the tool command language/toolkit or as it is best known Tcl/Tk, scripting language (Ousterhout 1994). The GUI provides the user with a single interface to set up, modify, and visualize a run of the ZEUS code. By utilizing the Tcl/Tk language, the GUI can easily be modified as ZEUS evolves.

The interface allows users to load, modify, and save the Fortran input files as opposed to the error-prone method of editing input files in a text editor. In addition, the user can save a custom file that also captures the state of the GUI. This allows the user the flexibility of running the code entirely from the GUI while not limiting users who wish to run from the command line. The ZEUS GUI introduces an archive method that allows users to load, save, and browse a custom archive file. This contains inputs, results, and comments for various runs of ZEUS. This feature allows users to easily compare results or duplicate previous runs.

Each input parameter can be modified via the interface by manipulation of scales and radio boxes. Radio box is a term used to describe a graphical means by which mutually exclusive options for a given input are presented to the user. These "widgets" help to ensure the validity of input parameters by disallowing values outside acceptable range. The scales and radio boxes are contained in a novel "hypertext" widget that combines the input widgets with explanation text. This provides the novice user with complete instructions but does not encumber the experienced user. Figure 2 depicts the GUI and shows some widgets employed to aid users with their inputs. More conventional, full help facilities are also included.

Finally, the interface allows the user to execute the FAST visualization program. The ZEUS GUI is capable of building a custom script file for FAST, individualized to each case, once ZEUS has terminated. However, this feature is currently not implemented. Once a sufficiently broad database of cases has been accumulated to ensure that the scripts will perform correctly over an equally broad range of cases, the custom script feature will be made operational. At the present time, the GUI provides the user with a list of prewritten scripts, which can then be used to run FAST.

An element of interaction between the user and ZEUS should also be provided by the GUI. The GUI would not be very effective if no useful feedback were provided quickly to the user. This feedback could be in the form of various types of information. For the ZEUS GUI, the feedback comes in the form of a simulation of the ZEUS code execution. A solution grid is produced as a result of the simulation. The status of the solution grid is a good indicator of whether the input will direct the ZEUS code to perform as intended. The user can view the grid by using FAST. In most instances, the ZEUS simulation program, hereafter referred to as pseudo-ZEUS, requires less than a minute of execution time on a Silicon Graphics Incorporated (SGI) Indigo series workstation, while ZEUS requires tens of minutes to execute large cases. The GUI was written in such a way as to suggest to the user to take advantage of pseudo-ZEUS to check the input. If the input is not acceptable, the user may edit it using the GUI. Then, pseudo-ZEUS can be used again to check the input. These steps should be repeated until the user is satisfied that all the input is correct. Once this is done, the ZEUS code is run. Figure 3 outlines this iterative process.

The GUI provides a user-friendly means of controlling ZEUS and its auxiliary programs. However, it should be stressed again that the user must know how each code and file relate to one another. Figure 4

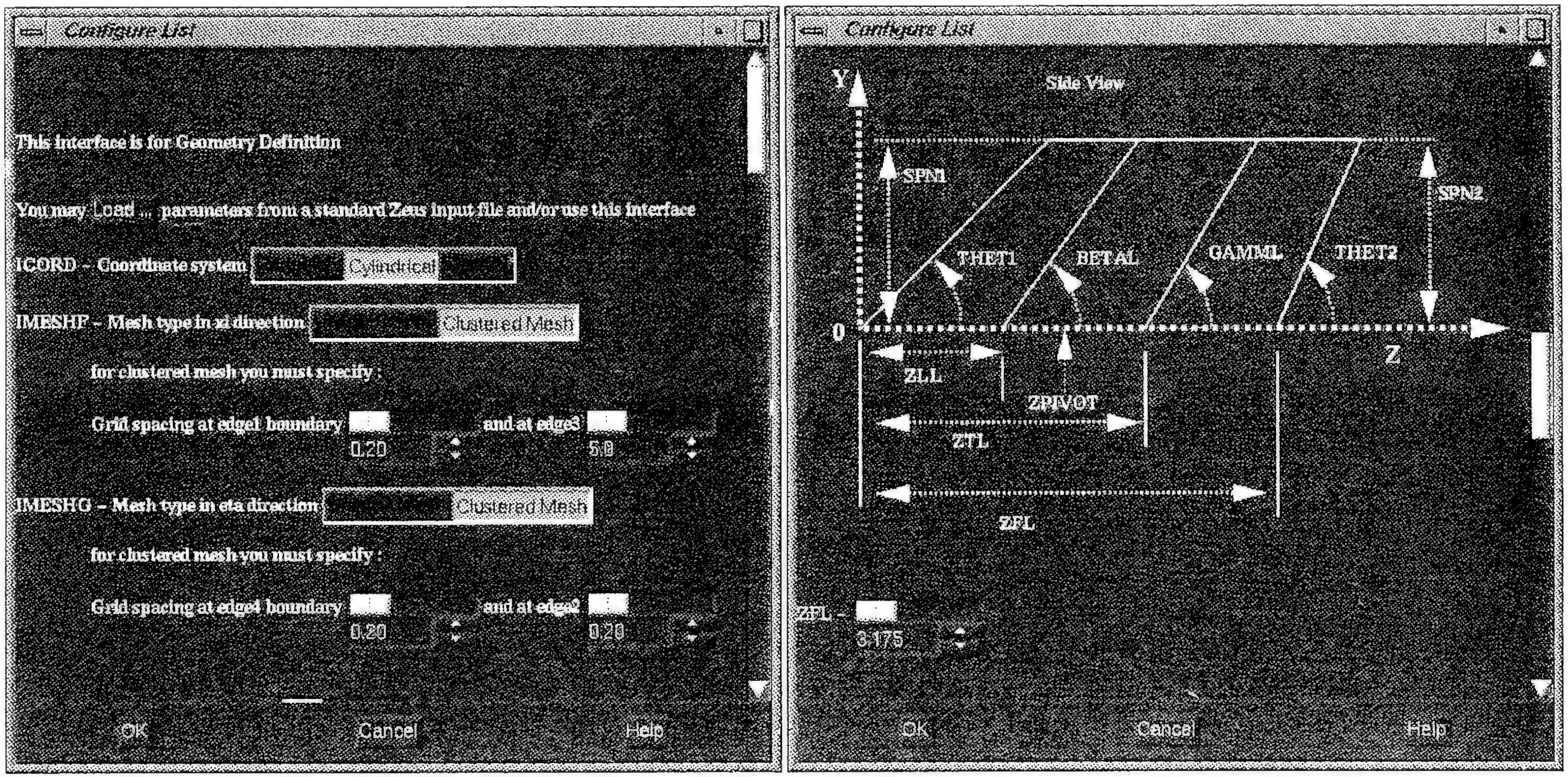

Figure 2. ZEUS graphical user interface.

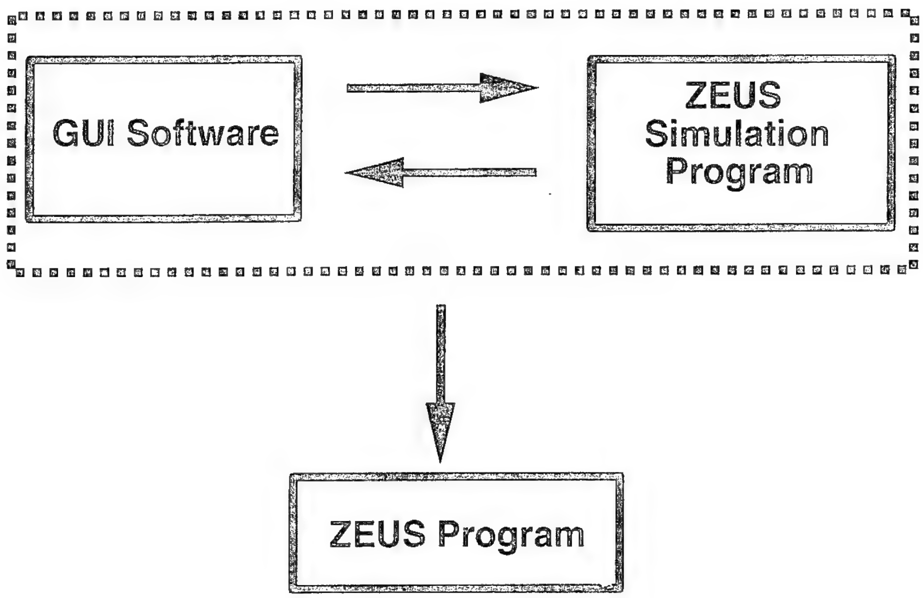

Figure 3. Graphical user interface diagram.

is a flowchart that shows the executable codes and pertinent files. There are other files created while running ZEUS and its auxiliary programs, but only the files necessary in executing ZEUS will be found in Figure 4. The user can access each executable from the GUI. The user can also manipulate the solution files from the GUI. The GUI can be used to manipulate and edit each input file: bc.inp, geo.inp, initc.inp, intcntrl.inp, outcntrl.inp, and zondim.inp with the exception of congeo.inp. The input file, congeo.inp, contains geometry information needed by the CONSTRT program to create the initial cone flow field solution plane. As indicated by Figure 4, congeo.inp is created by execution of the pseudo-ZEUS code. The user does not need direct access to congeo.inp. However, before the CONSTRT program can be executed, congeo.inp must be present. In a similar manner, the user does not have direct access to the geometry files, fedge2.p3d and fedge4.p3d. As indicated by Figure 4, executing the CONVERT program requires that fedge2.p3d and fedge4.p3d are present. The geometry files are created through the execution of either ZEUS or pseudo-ZEUS. All of the functions of the executables in Figure 4 have been documented earlier except that of FINCHECK. The FINCHECK program takes the information contained solely in the geometry file and returns an output file which can be read by FAST. The output file can be used to display a visual representation of a fin described by the geometry file input.

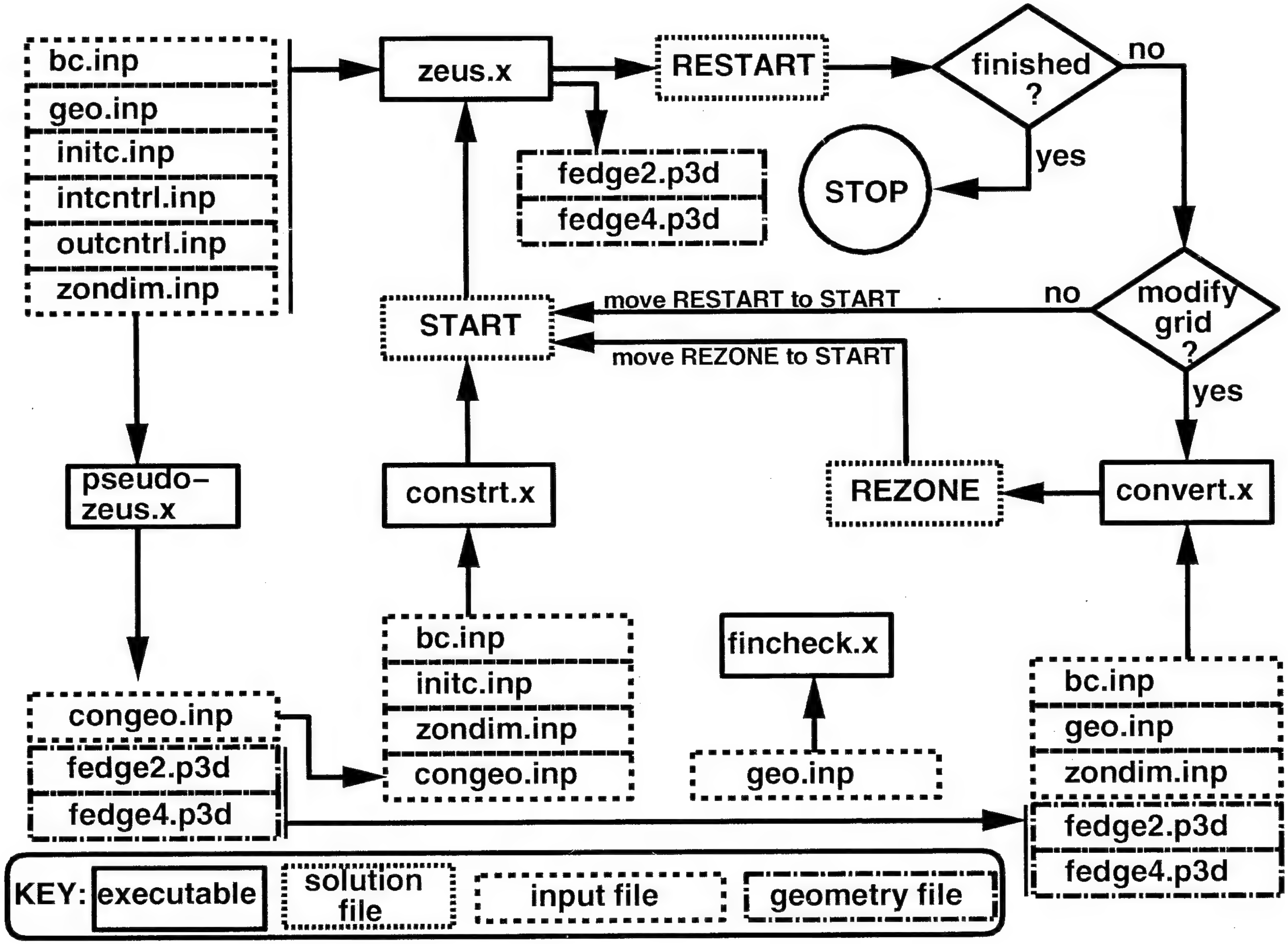

Figure 4. ZEUS graphical user interface flowchart.

4. VISUALIZATION

In order to make the GUI useful, some visualization must be provided. It was decided to couple an existing flow visualization program to the GUI rather than create one. As stated earlier, FAST was the chosen one. FAST was written and is now distributed by the National Aeronautics and Space Administration (NASA). FAST is fully interactive. The vast array of functions that FAST can perform should provide the user with adequate tools to analyze and visualize the data computed by ZEUS. A user manual is available if more information is needed about FAST (Walatka et al. 1991). FAST is very loosely coupled to the GUI. It would be a simple matter to replace FAST with another flow visualization program provided the new visualization program could read the PLOT3D formatted grid and solution files generated by ZEUS. The information coupling FAST to the GUI consists only of a path to the FAST executable and a path to prewritten FAST scripts. The FAST scripts contain commands that FAST executes. They have been written specifically to aid users in visualizing the grid and solution files generated by ZEUS and its auxiliary programs. The user needs only to choose and execute the script to visualize a grid or solution. For example, if the user wanted to see only the fin generated from the input, the user would choose the proper script name from a menu. FAST would then follow the commands in the script and display the fin visually. The user will not need any prior experience with FAST to visualize the data generated by ZEUS. However, the scripts do not exercise the full capabilities of FAST. A user should learn about FAST in order to exploit its many features. Figures 5 and 6 are examples of data generated by pseudo-ZEUS and ZEUS and respectively visualized by FAST. Figure 5 is a solution grid and fin diagnostic data generated by the execution of pseudo-ZEUS, while Figure 6 shows pressure contours on a body computed from a ZEUS solution.

5. APPLYING ZEUS

As currently configured, the ZEUS code is restricted to solving the flow field about finned axisymmetric missiles. Simple fin shapes with leading and trailing edge bevels can be modeled. Multiple sets of fins, including canards, can be modeled. A given fin set can have a variable number of fins although the ZEUS code is currently dimensioned to handle a maximum of eight. The cant angle of the fins can be varied. The attitude of the body can vary as well. The body pitch and yaw angles can be changed, and the body roll orientation can be modified as well.

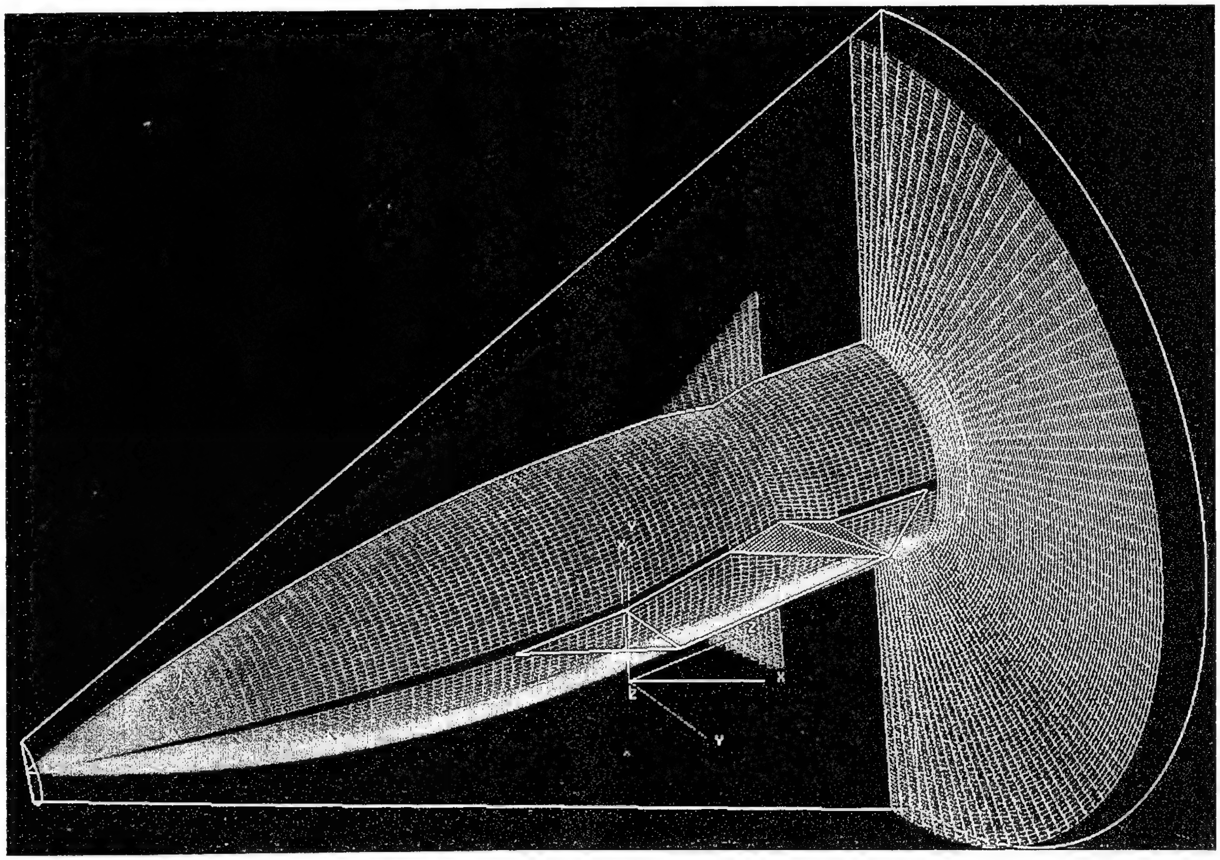

Figure 5. Pseudo-ZEUS grid.

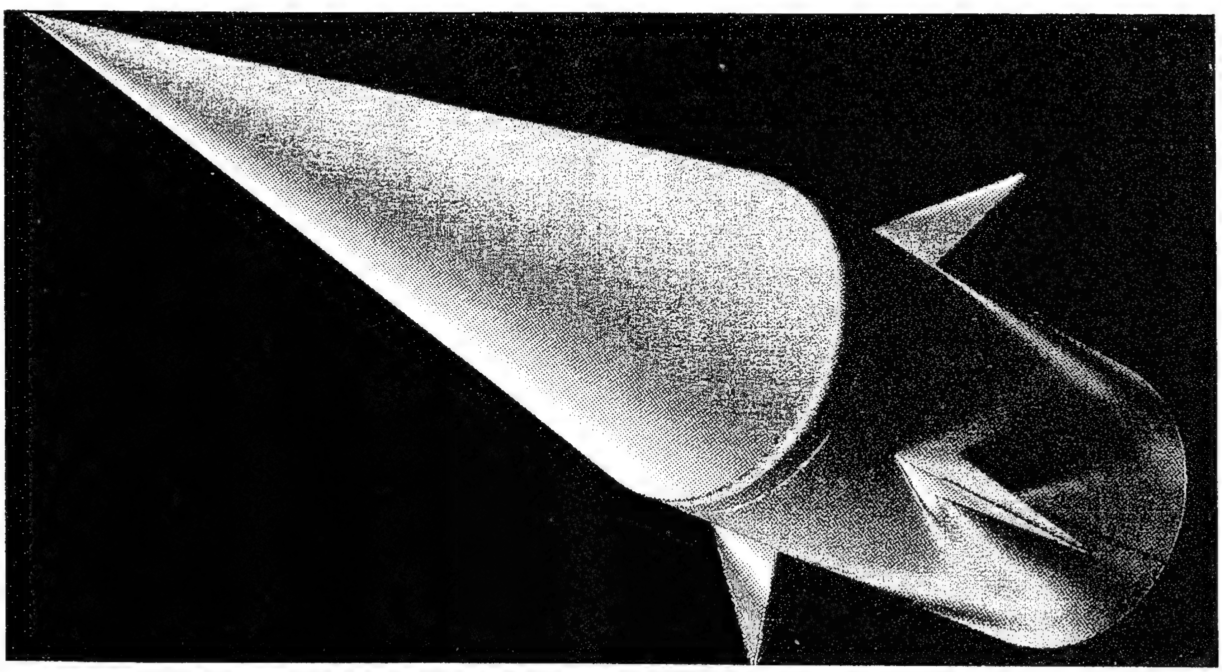

Figure 6. Surface pressure contours form ZEUS solution.

The following sections address some of the nomenclature, conventions, and variable names a user would find useful in applying the ZEUS code to missiles as it is currently configured. Experienced ZEUS users should be familiar with most of the application aspects discussed in the upcoming section. Although a significant number of variables have been added or modified, an effort was made to retain the basic language and style of the original noninteractive ZEUS code to facilitate a smooth transition, for experienced users, to the version documented in this report.

5.1 Coordinate System. The ZEUS code uses a general coordinate system for orientation of the body or overall missile, and a local coordinate system to describe the fin. The general coordinate system is shown in Figure 7, and the local coordinate system is shown in Figure 8. The figures follow the same nomenclature and conventions represented in the GUI. For example, in Figure 7, ALPHA is described. ALPHA is the variable that describes the angle of attack in the GUI as well. The fin cant angle, CNTANG, is described in the local fin coordinate system. ZPIVOT, another variable defined in the local fin coordinate system, indicates the point about which the fin will pivot. ZPIVOT also indicates the point of reference for determining where the fin is attached to the missile body. Figure 9 shows how the general coordinate system and the local coordinate system relate to one another. The variables, ZHINGE and RHINGE, dictate the position of the fin on the missile body.

5.2 Defining Zones. Zone definition primarily encompasses the topics of the number of zones, the number of points in each zone, and the circumferential area each zone will occupy. ZEUS is capable of generating multiple zone grids. For ZEUS, zone boundaries may act as fin plane locations and grid-clustering control points. Figure 10 shows the variables necessary to define a zone, along with associated naming conventions. For example, zone 4 shows the naming convention for zone edges. Figure 10 represents a grid that has four zones. The variable IZN controls the number of zones. The number of zones dictates the number of default fin surfaces. This is because fin surfaces are modeled on the first and last planes in the circumferential (or eta) direction in each zone. Also note that when IASYM is zero, no pitch-plane symmetry is assumed and the grid covers the full 360° of space. When IASYM is one, pitch-plane symmetry is assumed, and the grid covers only 180° of space. The total number of points in the normal (or xi) direction is determined by NA. The total number of points in the eta direction is determined by MA. The user must also specify the number of points in the eta direction for each zone. This information is contained in the array MAZ. The sum of the values in array MAZ must be equal to MA.

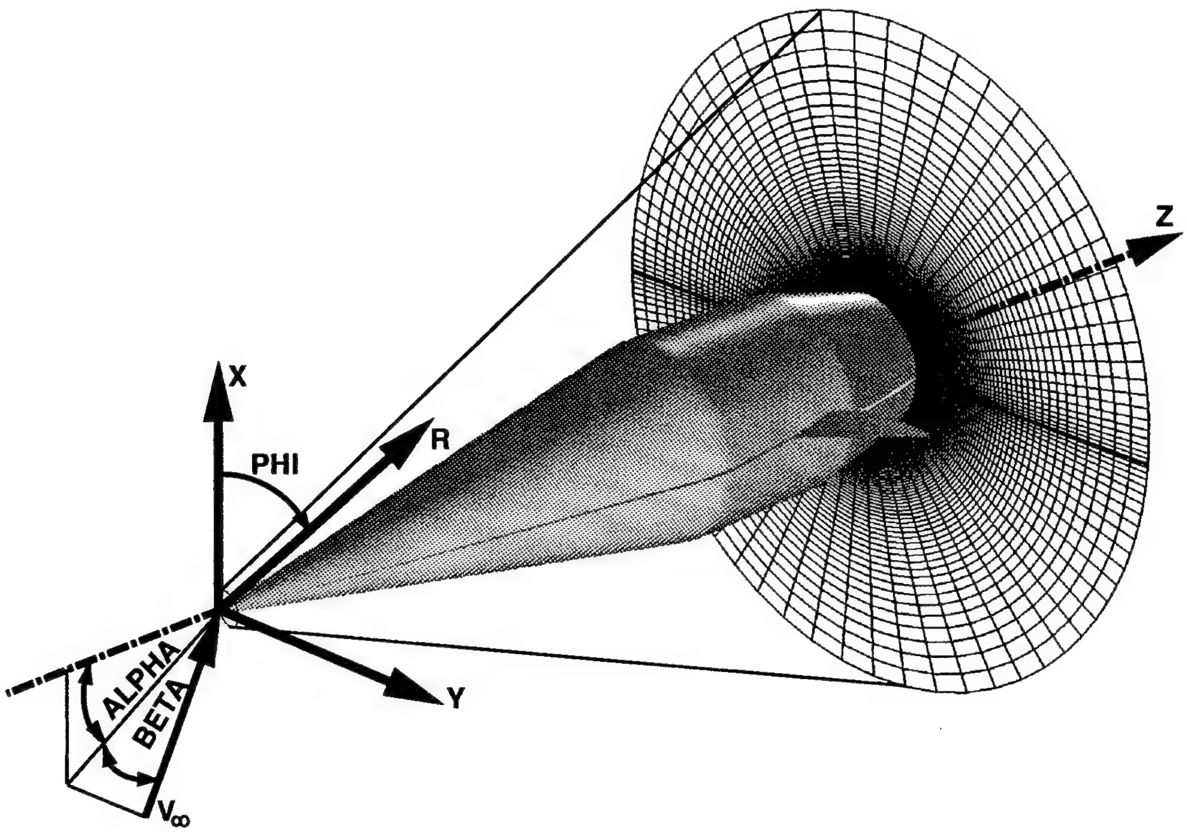

Figure 7. ZEUS general coordinate system.

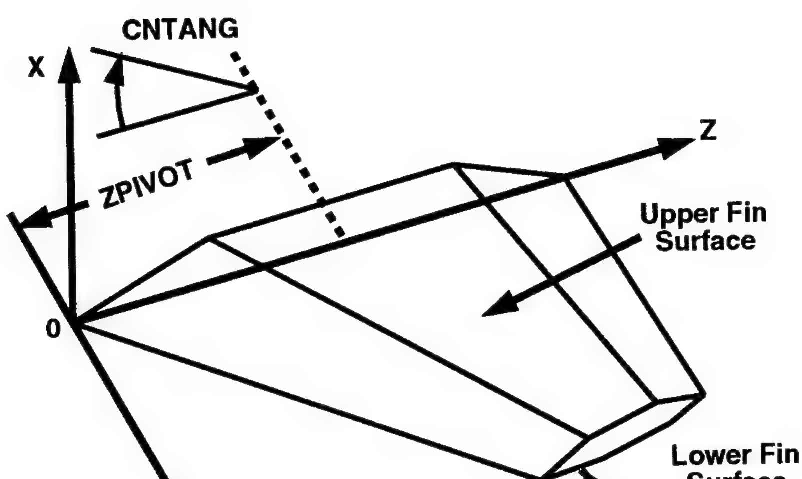

Figure 8. Local fin coordinate system.

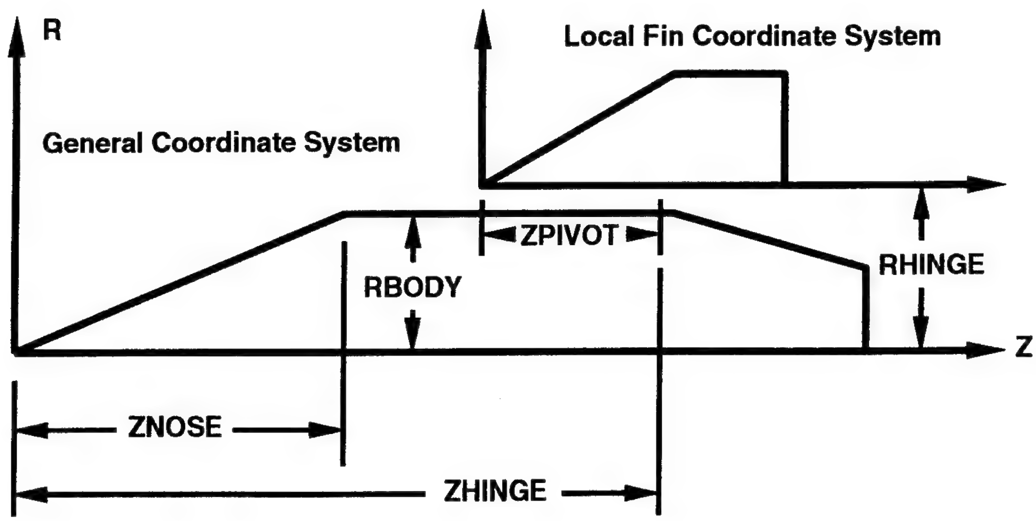

Figure 9. General coordinate system and local coordinate system.

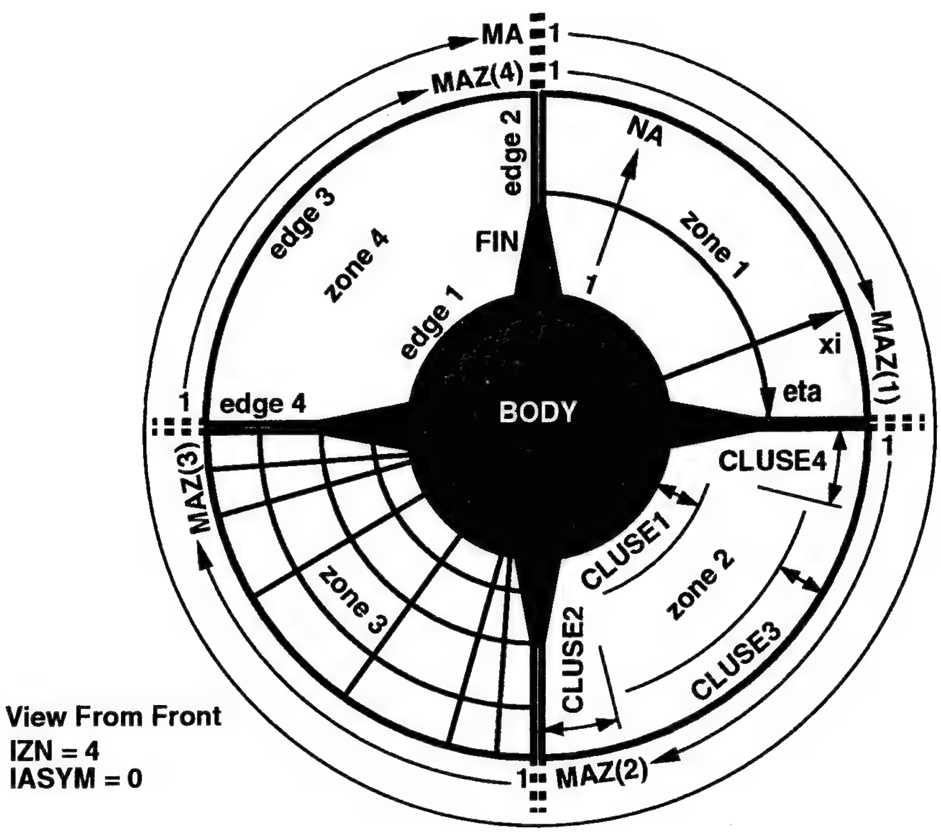

Figure 10. Zone definition.

By default, the zones cover an equal circumferential area. Thus, by default, the fins are equally spaced apart. For example, if the total grid covers 360° of space and there are four zones, then each zone will cover 90° of space and the fins will be spaced 90° apart. If the user-defined zone spacing option is chosen, then the circumferential position of each fin can be determined. Figure 11 represents the user-defined zone spacing option being exercised for a missile with three fins. The variable INPANG determines if the user-defined zone spacing is used. When INPANG is zero, the fins will be placed equidistantly around the body and the user-defined zone spacing is not used. When the user-defined spacing is opted for, INPANG is one, and the circumferential angle covered by each zone must be supplied to the array ANGMAZ. Further adjustment of the fin position can be made through the variable PHIBOD. Figure 12 demonstrates how the variable PHIBOD can be used to position the fins. PHIBOD is helpful for simulating banking of maneuvering missiles. Use of the variable PHIBOD is only valid when there is no assumption of pitch-plane symmetry.

Grid clustering is another aspect of how zone definition affects the fins. In order to model the fin geometry accurately, it is recommended that clustering be used in the circumferential direction. This is especially true of thin fins. In Figure 10, one can see the naming convention for the variables that control clustering—CLUSE1, CLUSE2, CLUSE3, and CLUSE4. CLUSE1 controls the point spacing at edge1, CLUSE2 controls the point spacing at edge2, etc. The clustering values are actually percentages of arc length. The clustering values are not needed if the uniform mesh option is chosen.

5.3 Boundary Conditions. As stated earlier, the ZEUS code is restricted to solving the flow field about axisymmetric missiles. With this limitation in mind, default boundary conditions have been encoded into the current version of the ZEUS code to expedite the setup of the solution. Figure 13 shows the default boundary conditions for a solution employing two zones and pitch-plane symmetry. A surface boundary condition is automatically designated at the first plane in the xi direction. This represents the missile body. A surface boundary condition is designated for the grid points occupying the region defined by the geometry inputs as part of the fin on the first and last eta planes of each zone. Those points on the first and last eta planes of each zone not contained in the fin region are defined as interior points if there is no pitch-plane symmetry. If there is pitch-plane symmetry, as shown in Figure 13, then a pitch-plane symmetry boundary condition is applied to grid points on the plane of symmetry, or the Y equals zero plane, not contained in the fin region. The grid points to which the symmetry boundary condition is applied can also be described as the set of points on the first eta plane of the first zone and the last eta plane of the last zone that are not contained in the area designated as part of the fin. Those grid points

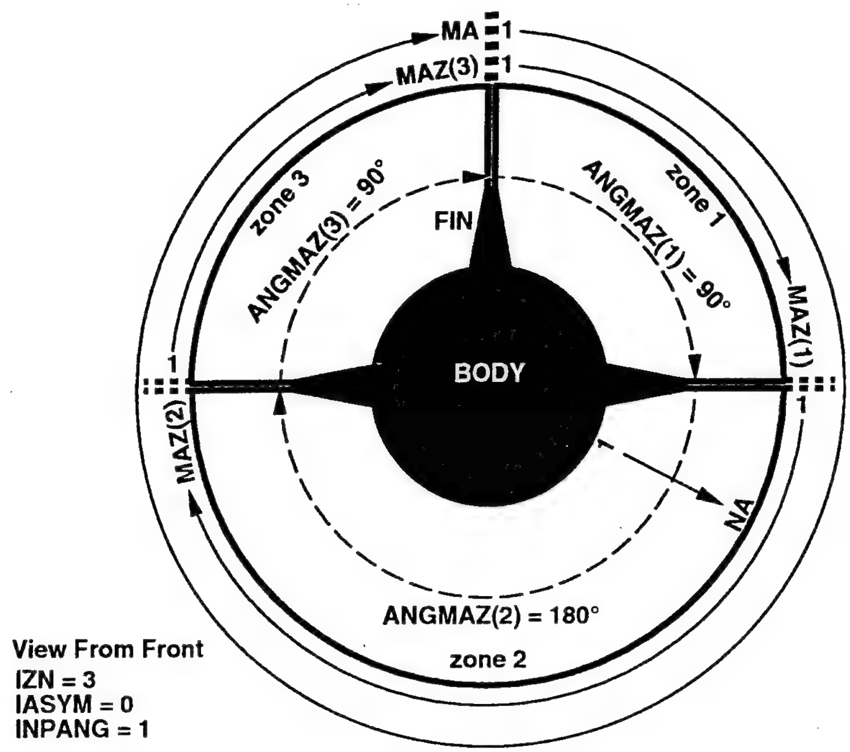

Figure 11. User-defined zone spacing.

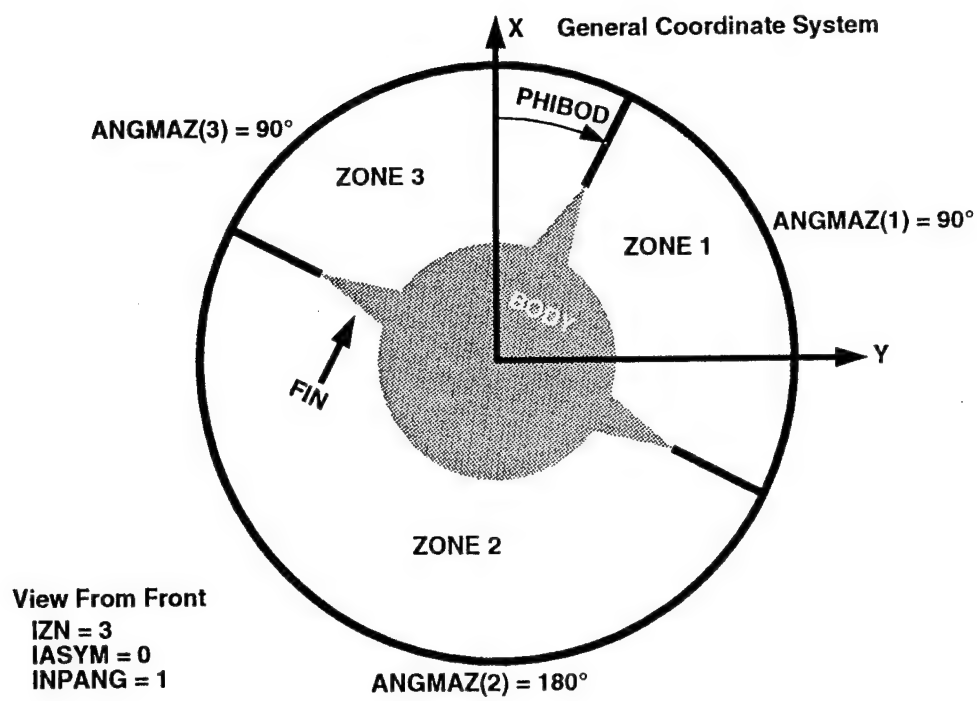

Figure 12. Body rotation angle.

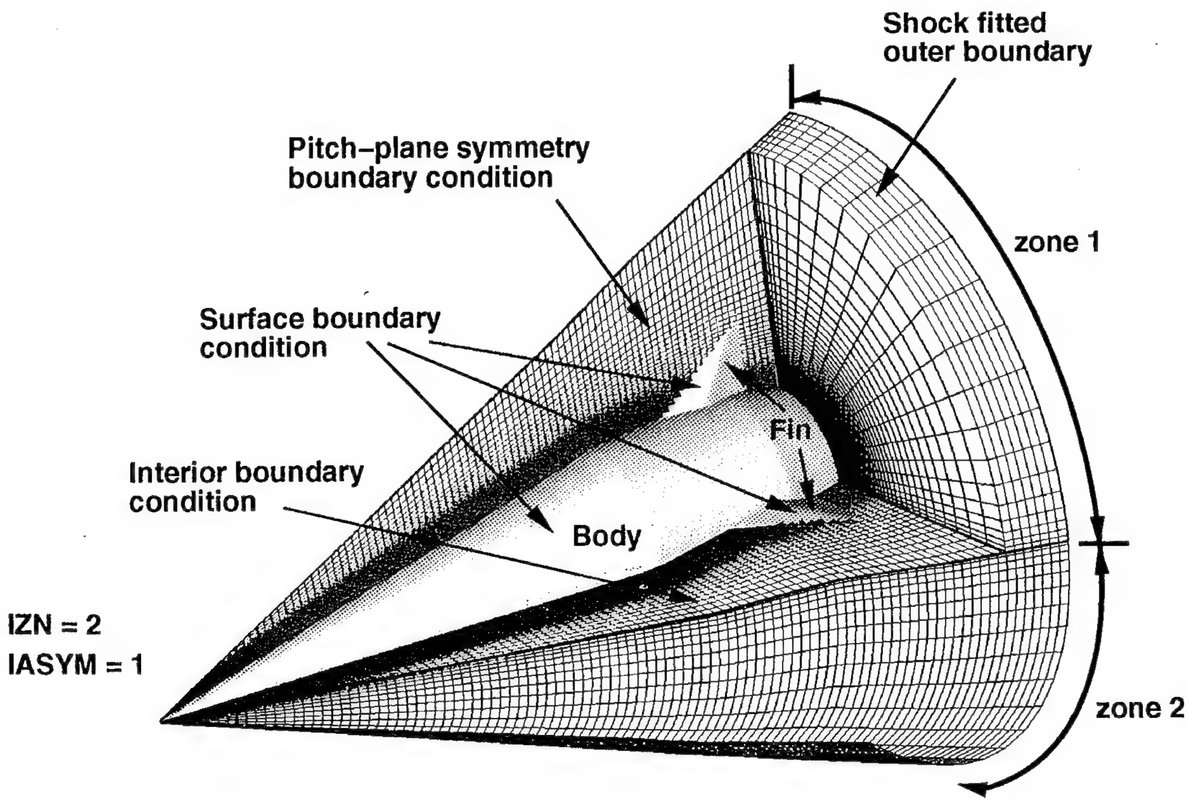

Figure 13. ZEUS boundary conditions.

on eta planes that are adjacent to another zone and are not designated as part of the fin surface are designated as interior points. The last plane in the xi direction, or the outer boundary, is shock-fitted. Although not every missile configuration can be anticipated, it is expected that the default boundary conditions will need no modification for most cases. In the event the user needs to modify the boundary conditions, provisions have been made to do so through the use of the GUI.

5.4 Geometry. The current set of geometry input has been chosen to describe finned axisymmetric missiles. Although the user must provide a number of input variables, defining the geometry is fairly simple. Since the variables used to describe the body and fin are defined in different coordinate systems, a natural subdivision for addressing the geometry input has been provided. The variables for describing the body must always be defined, while the variables used to describe the fin are optional. If a fin is defined, the parameters relating the general coordinate system used to describe the body and the local coordinate system used to describe the fin must be defined as well. These parameters—ZPIVOT, ZHINGE, and RHINGE—were mentioned earlier and are shown in Figure 9.

Figure 14 shows the variables used to model the body. Only ZNOSE and RBODY are required to define a simple body. In Figure 14, ZNOSE defines the length of the nose while RBODY describes the radius of the body at z equal to ZNOSE. Figure 14 shows the nose to be a cone. However, the GUI provides the user with the option of choosing the nose to be an ogive. Also shown in Figure 14 are body description points. Body description points are optional. If no body description points are defined, then the body is assumed to be a cylinder with the body radius equal to RBODY. To describe a body with a varying radius, the user must supply the number of body description points to be used, NBDPNT, and the z and r coordinates for each point. The z and r coordinates are stored in the arrays ZBDPNT and RBDPNT, respectively.

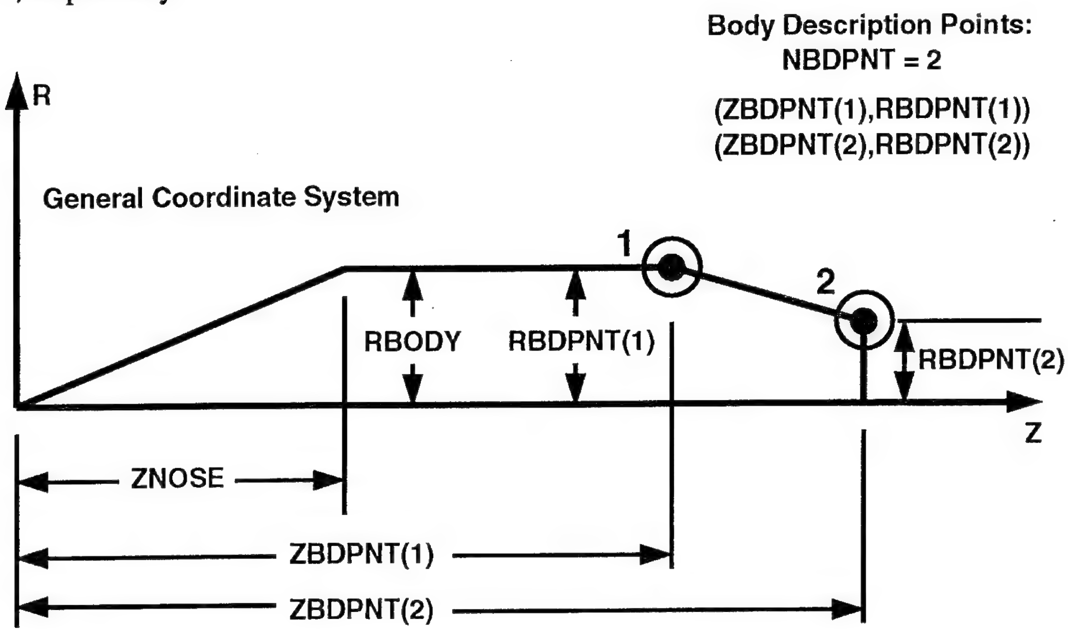

Figure 14. Body modeling.

As mentioned earlier, the input variables describing the fin have their own local coordinate system, which can be seen in Figure 6. Figure 15 shows the input needed to describe the fin. The fin has an upper and a lower surface. Figure 15 shows only one side view of the fin surface. For the variables BETAL, GAMML, ZLL, and ZTL, there is a corresponding set of variables that describe the opposite surface. These variables are BETAU, GAMMU, ZLU, and ZTU, respectively. The fin description variables were designed to provide the user with a number of fin shape choices while still conforming to the geometric limitations of the ZEUS solution algorithm. Figures 16a, 16b, 16c, and 16d show examples of fins that can be defined with the input. Figure 17 depicts another aspect of fin modeling, the fin-body juncture. The variable, IFNGAP, controls how the fin is attached to the body. If no fin gap is modeled, IFNGAP is set to zero and the base of the fin conforms to the shape of the body. If a fin gap is to be

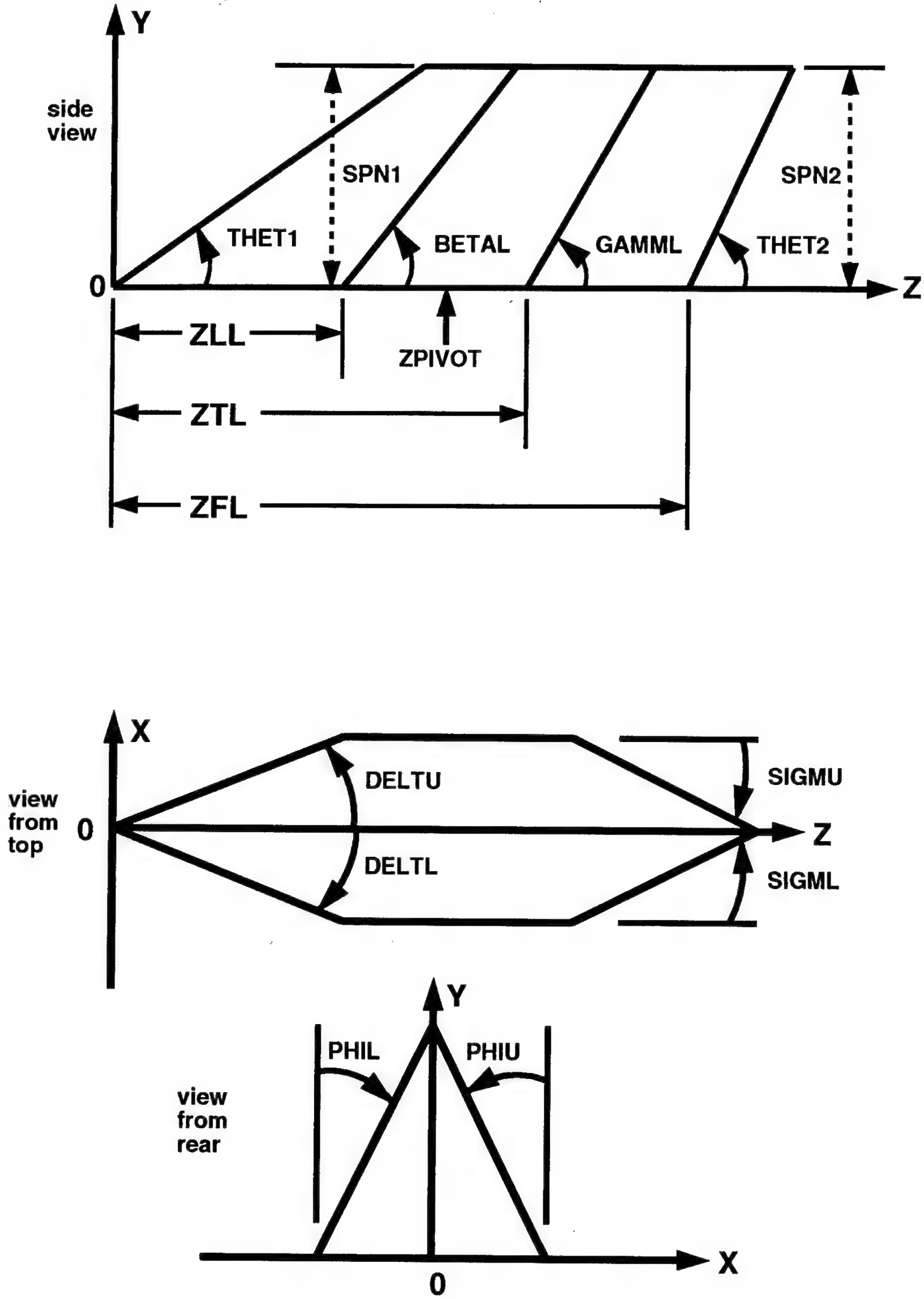

Figure 15. ZEUS fin input.

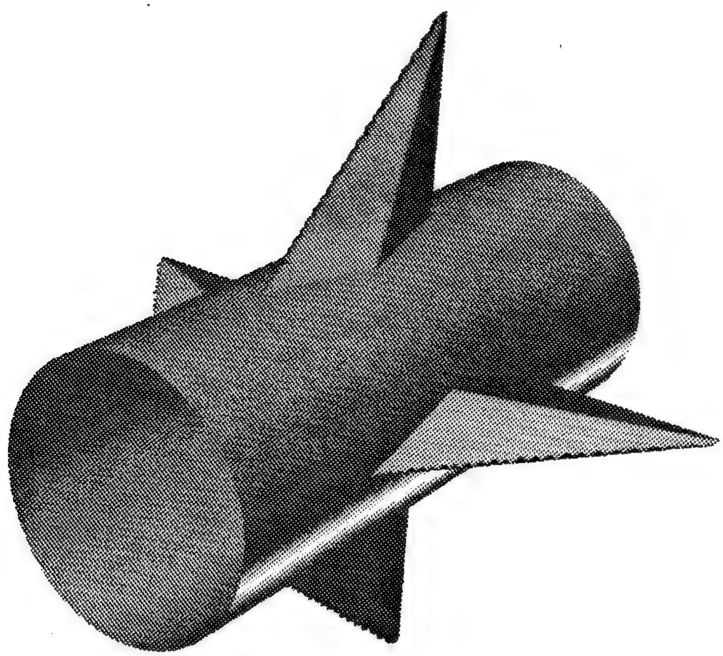

Figure 16a. ZEUS fin example

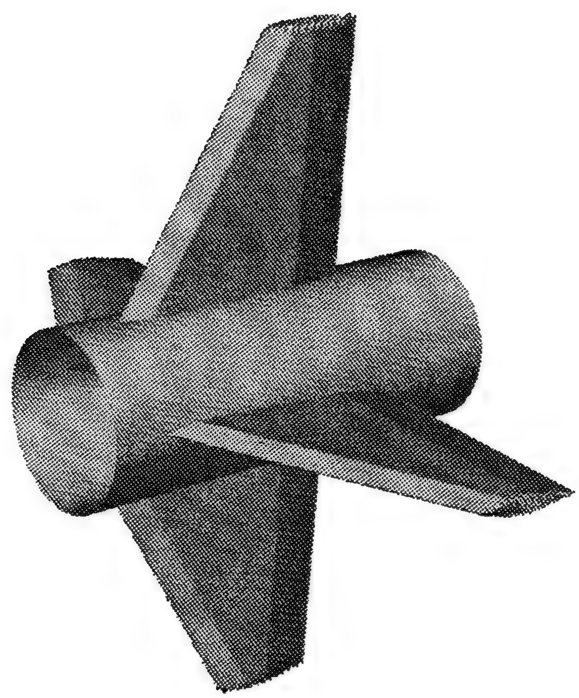

Figure 16b. ZEUS fin example.

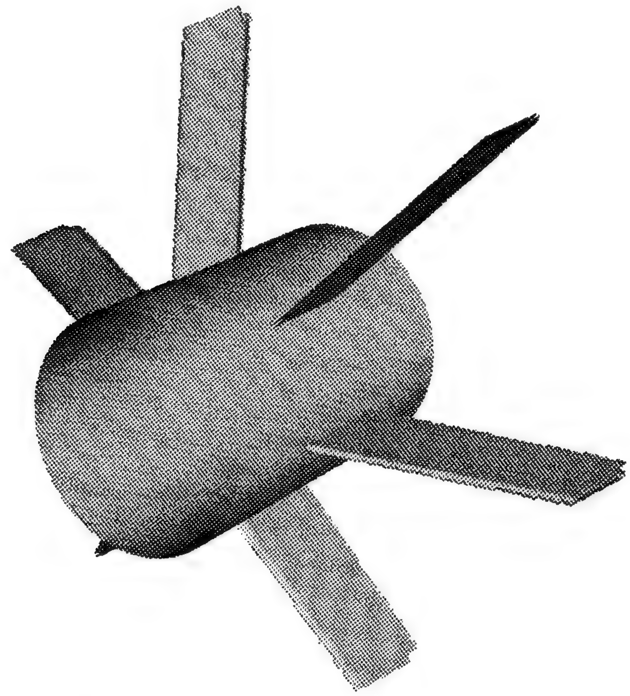

Figure 16c. ZEUS fin example.

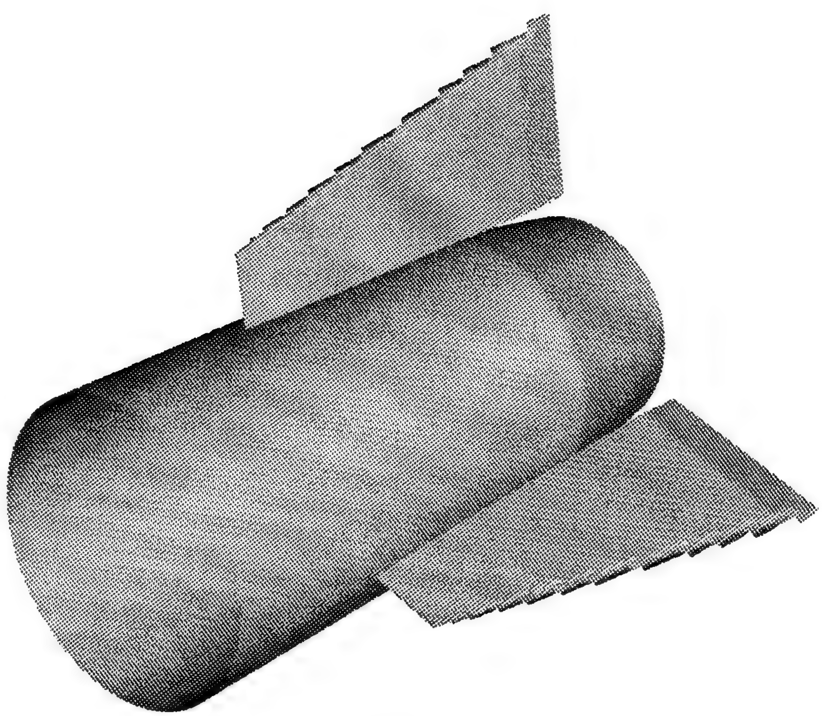

Figure 16d. ZEUS fin example.

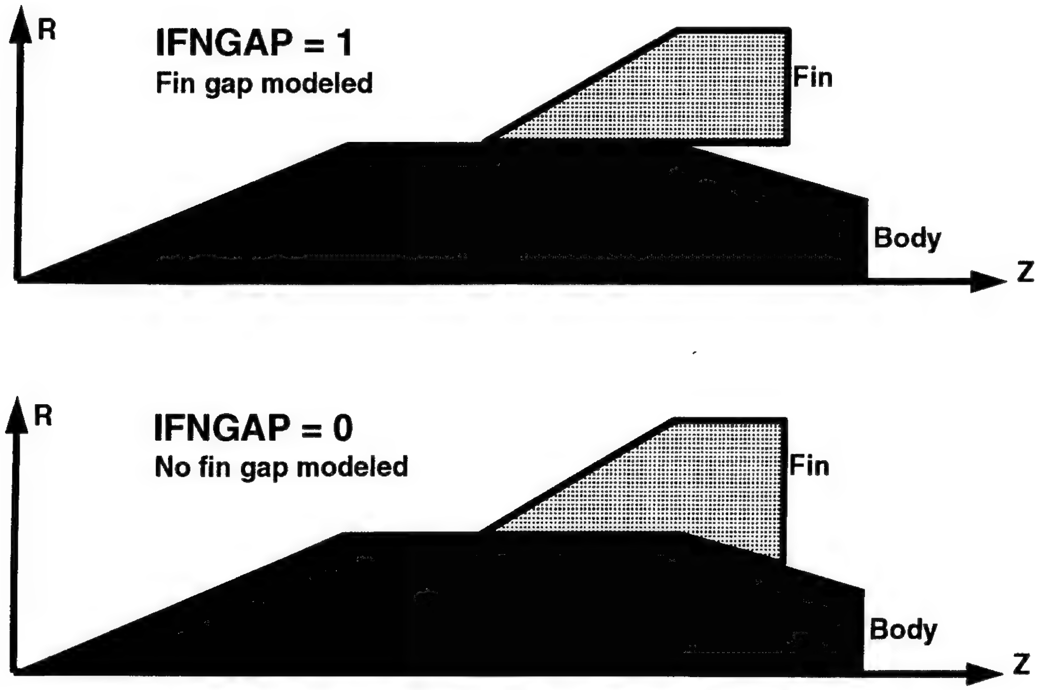

Figure 17. Fin gap modeling.

modeled, then IFNGAP is set to one and the fin root is kept at a constant radius. Figure 16d is an example of a fin modeled with a gap. The distance between the fin and the body can then be controlled by the variable RHINGE, which can be found in Figure 9. Fin gap modeling should be used only with very thin fins.

When viewing the pseudo-ZEUS-generated grid file, the user should carefully inspect the fin region. Figure 5 shows a grid generated by the pseudo-ZEUS program. FAST, directed by one of the prewritten scripts, was used for the visualization. The shaded region represents the actual coordinates the user defined through the fin variable inputs. The outline of the shaded region represents auxiliary information needed by ZEUS to smoothly blend the grid into the leading and trailing edges of the fin. The fins generated by the pseudo-ZEUS program appear as a grid. The user should always check for a good match between the actual coordinates and the fins generated by pseudo-ZEUS. Another potential problem to watch for is grid discontinuity in the fin region. ZEUS will not be able to compute a solution if the grid is discontinuous. As stated earlier, the CONVERT program interpolates a modified grid and solution based on grid refinement inputs. However, CONVERT should not be applied in an area where a fin is present. The region in which CONVERT cannot be applied is represented by the shaded fin.

6. SUMMARY

It is hoped that the GUI designed for ZEUS will provide a useful tool for design and analysis of axisymmetric missile configurations. Although not all of the ZEUS variables addressable through the GUI were discussed in this report, enough of them were presented to give the potential user an idea of the GUI's and ZEUS's capabilities. A complete list of the ZEUS input variables can be found in the Appendix. Some improvements and/or adjustments will be made in the future. For example, the restriction to axisymmetric bodies is a glaring limitation. A version of ZEUS may be written to handle some cases of nonaxisymmetric bodies. It is difficult to foresee a single version of ZEUS that will be applicable to all cases. A single all-encompassing version of the ZEUS code would probably be too bulky, too complex, and too slow. It is more likely that a series of ZEUS codes will be written, and each will handle a specific subset of cases. The GUI would direct the user to choose the version of ZEUS that is applicable to their case and display the pertinent variables needed to execute the chosen version. This approach has several advantages. It allows for modular upgrades and/or modifications, and it allows for the capabilities of the ZEUS GUI to grow while still maintaining some measure of user friendliness and simplicity. Most importantly, it allows the ZEUS GUI to be a useful tool not only today but in the future as well.

INTENTIONALLY LEFT BLANK.

7. REFERENCES

Ousterhout, J. "Tcl and the Tk Toolkit." Reading, MA: Addison-Wesley Publishing Company, September 1994.

Walatka, P., T. Plessel, R. McCabe, J. Clucas, and P. Elson. "FAST User's Manual." RND 91-011, National Aeronautics and Space Administration, Ames Research Center, Moffet Field, CA, December 1991.

Wardlaw, A., and S. Davis. "A Second Order Godunov Method For Supersonic Tactical Missiles." NSWC TR 86-506, Naval Surface Weapons Center, Silver Spring, MD, December 1986.

Wardlaw, A., and F. Priolo. "Applying the ZEUS Code." NSWC TR 86-508, Naval Surface Weapons Center, Silver Spring, MD, December 1986.

INTENTIONALLY LEFT BLANK.

APPENDIX:

INPUT FILE VARIABLES FOR ZEUS

INTENTIONALLY LEFT BLANK.

The following is a list of variables and a brief description of their purpose. The variables are grouped according to their general function. This is also the way they are organized in the GUI. For example, variables listed under initial conditions in this appendix will appear under the same heading in the GUI. The GUI will store all variables under the same heading in one file. The variables listed under initial conditions, integration control, zone dimensions, boundary conditions, geometry definition, and output control will be stored in the input files initc.inp, intcntrl.inp, zondim.inp, bc.inp, geo.inp, and outcntrl.inp, respectively. These files are listed in the flowchart pictured in Figure 4. The variables listed in the appendix are also written in the order and the format in which they would appear within their respective files. If necessary, it is very easy to manually edit the input files. Most of the variables retain their original functions from previously written versions of the noninteractive ZEUS code, and their definitions were obtained from existing text. Some variables listed in the appendix were not discussed earlier. However, each variable listed can be accessed through the GUI. A review of Wardlaw's work is recommended and may provide useful insight on properly utilizing the functions controlled by these variables (Wardlaw and Davis 1986; Wardlaw and Priolo 1986).

INTENTIONALLY LEFT BLANK.

VARIABLE DESCRIPTION

*****INITIAL CONDITIONS*****

ALPHA Angle of attack (degrees)

BETA Angle of yaw (degrees)

XMINF Mach number

PINF Pressure

DINF Density

*****INTEGRATION CONTROL*****

IMOD 0,1 for new start or restart, respectively.

ZSTART Begin computation at Z = ZSTART

ZETAEND Terminate calculation if ZETA > ZETAEND.

KEND Terminate after KEND steps.

FCFL Step size safety factor (1.0 > FCFL > 0.0). Typically use 0.9.

XKI Interior point limiting constant (2.0 > XKI > 0.0). Typically use 1.0.

IAPR 0,1 for approximate or complete Riemann Problem. Typically use 0.

DFAC In case of a local wall Mach number which is too small to allow the flow to be turned parallel to the wall, the turn angle is multiplied by the constant DFAC (1.0 > DFAC > 0.0).

IVIS 0,1,2 for no modeling, clipping, and forced separation. Clipping and forced separation can only be used in conjunction with cylindrical coordinates. If IVIS = 2, specify the number of separation lines (NSEP).

>>>>> list variable NSEP

NSEP

<<<<<end list

For each separation line specify:

ISSIDE - 0 separation line located between 0 and 180 degrees,
1 separation lines located between 180 and 360 degrees.

ISEP - number of points used to define the separation line.

ZSSEP - Zeta value at which separation is started

ZESEP - Zeta value at which separation is terminated

PHICD - separation angle PHIC (degrees)

PHIAD - separation angle PHIA (degrees)

BETACD - separation angle BETAC (degrees)

BETAAD - separation angle BETAA (degrees)

>>>>> start list of [ISSIDE,ISEP,ZSSEP,ZESEP,PHICD,PHIAD,BETACD,BETAAD]

ISSIDE,ISEP,ZSSEP,ZESEP,PHICD,PHIAD,BETACD,BETAAD

<<<<< end list

A list of ISEP pairs of points (SEPZ,SEPP) describing the zeta and phi coordinates of the separation line. This list should start at the most forward section and move aft. Start the list for each separation line on a separate line and list in same order as used in the above data.

>>>>> start list of [SEPZ,SEPP]

SEPZ(ISEP),SEPP(ISEP)

<<<<< end list

*****ZONE DIMENSIONS*****

IZN Number of zones.

NA Number of cells in xi direction (i.e., between edges 1 and 3)

MA Number of cells in eta direction (i.e., between edge 4 of zone 1 and edge2 of zone IZN)

List the number of M or eta planes in each zone, starting from zone 1 and ending at zone MA.

>>>>>start list of MAZ for each zone

MAZ(IZN)

<<<<<end list

INPANG 0 - equidistant zone spacing; 1 - user defined zone spacing

If INPANG = 1 must list the total angle (degrees), that each zone is to cover, starting from zone 1 and ending at zone IZN.

>>>>>start list of ANGMAZ for each zone

ANGMAZ(IZN) (degrees)

<<<<<end list

*****BOUNDARY CONDITIONS*****

IASYM 0 - no pitch plane symmetry; 1 - pitch plane symmetry.

NXKE Number of edges at which default limiter setting won't be used.

Default settings are 2 on edges 1 or 3 and 0 on edges 2 or 4. For each edge at which the default limiter is nonstandard, add a line containing the following information: zone number (KEZ), edge number (KEE), and limiter value (XKE)

>>>>>start list of [KEZ,KEE,XKE]

KEZ(NXKE),KEE(NXKE),XKE(NXKE)

<<<<<end list

NSUR Number of edges not featuring default surface types. Default values are 1 on edges 1 and 3 and 0 on edge 2 and 4. Program will automatically account a fitted shock on edge 3 or fin surfaces on edges 2 or 4.

For each edge at which the surface type is not of default, add a line that specifies the zone number(KSURZ), edge number(KSURE), and surface type (KSUR).

>>>>>start list of [KSURZ,KSURE,KSUR]

KSURZ(NSUR),ZSURE(NSUR),ZSUR(NSUR)

<<<<<end list

ISHOCK Set to 1 if edge 3 is to be fitted by the calculation (either as a shock or sonic line); otherwise set to 0. ISHOCK = 1 is only valid for ICORD =1 or 2. If edge 3 is to be fitted, the r, (RSHOCK) and phi (cylindrical coordinates) or s, tau (elliptic coordinates) of the shock location on the first M plane and the last M plane (PHI1SH, PHI2SH) must be defined.

>>>>>Start list of [RSHOCK,PHI1SH,PHI2SH]

RSHOCK,PHI1SH,PHI2SH

<<<<<end list

*****GEOMETRY DEFINITION*****

ICORD 0,1,2 for cartesian, cylindrical, or elliptic coordinates

IMESHF 0 for uniform mesh in xi direction. 1 for clustered mesh.
if IMESHF = 1 must provide:
CLUSE1 grid spacing at edge1 boundary
CLUSE3 grid spacing at edge3 boundary

IMESHG 0 for uniform mesh in eta direction. 1 for clustered mesh.
if IMESHG = 1 must provide:
CLUSE4 grid spacing at edge4 boundary
CLUSE2 grid spacing at edge2 boundary

RBODY body cylinder radius

ZNOSE nose length

NOSTYP 0 = cone, 1 = ogive

NBDPNT number of additional body description points
if NBDPNT > 0 must provide NBDPNT Z,R coordinates along body:
ZBDPNT(NBDPNT), RBDPNT(NBDPNT)

PHIBOD body rotation angle about the Z-axis (degrees)

IFINS 0 = no fins, 1 = fins
if IFINS = 1 must provide fin geometry inputs:

ZHINGE Z-ordinate where zpivot is attached to projectile body

RHINGE R-ordinate where zpivot is attached to projectile body

The following inputs are made in a local coordinate frame at which the Z-ordinate of the leading edge at the root is 0. See Figure 5 for reference.

ZPIVOT Z-ordinate about which the fin is rotated when the fin cant angle does not equal zero. Also used as the reference point for attaching the fin to the body.

CNTANG fin cant angle (degrees)

ZFL see Figure 14 for reference

ZLU see Figure 14 for reference
ZLL see Figure 14 for reference
ZTU see Figure 14 for reference
ZTL see Figure 14 for reference
SPN1 see Figure 14 for reference
SPN2 see Figure 14 for reference
THET1 see Figure 14 for reference (degrees)
THET2 see Figure 14 for reference (degrees)
BETAU see Figure 14 for reference (degrees)
BETAL see Figure 14 for reference (degrees)
GAMMU see Figure 14 for reference (degrees)
GAMML see Figure 14 for reference (degrees)
DELTU see Figure 14 for reference (degrees)
DELTL see Figure 14 for reference (degrees)
SIGMU see Figure 14 for reference (degrees)
SIGML see Figure 14 for reference (degrees)
PHIU see Figure 14 for reference (degrees)
PHIL see Figure 14 for reference (degrees)

*****OUTPUT CONTROL*****

IEFORCE 0, 1 don't, do print force and moments for individual edges.

AREF Reference area used in calculating force and moment coefficients.

XLREF Reference length used in calculating moment coefficients and center-of-pressure.

IPLT3D 0 = do not write PLOT3D file, 1 = write PLOT3D file
if IPLT3D = 1 must provide:
INCP3D increment used to write Z-stations in file

IOZEUS 0 = no pressure or force summary, 1 = print pressure & force summary
if IOZEUS = 1 must provide:
---for Printer---
IPRINT Print crossflow plane if step number is evenly divided by IPRINT

NSKIP Print n or xi planes which are evenly divisible by NSKIP

MSKIP Print m or eta planes which are evenly divisible by MSKIP

ISKIP Print step size if step number is evenly divided by ISKIP

---for PLOTZA/Force-Pressure Summary---
JSPPR In default mode, only edge 1 of each zone will be written to PLOTZA. JSPPR is the number of additional edges which are not to be written in the default manner. For each edge added or deleted from this print list, include a line which specifies zone number (JSPZ), edge number (JSPE), and print code (JSP). JSP = 1 will write this zone edge to PLOTZA, 0 will not.
>>>>>start list of [JSPZ,JSPE,JSP]
JSPZ(JSPPR),JSPE(JSPPR),JSP(JSPPR)

<<<<<end list

KSKIP Write surface properties on selected edges to PLOTZA if STEP NUMBER is evenly divisible by KSKIP

NPRT On summary sheet, print surface pressures of edges 2 or 4 if n < NPRT.

DELZA Write to PLOTZA if (ZETA - ZETA at last write) > DELZA

---for PLOTZC (Plot file)---

IPLOT Write to PLOTZC if STEP NUMBER is evenly divisible by IPLOT.

DELZC Write to PLOTZC if (ZETA - ZETA at last write) > DELZC

IPLOTN Number of target Z stations (ZPLOT) at which PLOTZC will be written. If IPLOTN > 0, include a list of these stations on the next line (maximum of 20). Stations must be listed in ascending order.

>>>>>start ZTARGET list

ZPLOT(IPLOTN)

<<<<<end list

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFO CTR ATTN DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL OP SD TA 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL OP SD TL 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL OP SD TP 2800 POWDER MILL RD ADELPHI MD 20783-1145
	ABERDEEN PROVING GROUND
2	DIR USARL ATTN AMSRL OP AP L (305)

NO. OF COPIES	ORGANIZATION
6	COMMANDER US ARMY MISSILE COMMAND ATTN AMSMI RD SS AT VAUGHN REDSTONE ARSENAL AL 35898-5252
1	COMMANDER US NAVAL SRFCE WARFRE CTR ATTN CODE B44 A WARDLAW SILVER SPRING MD 20903-5640
1	COMMANDER US NAVAL SRFCE WARFRE CTR ATTN CODE G04 FRANK MOORE WEAPONS SYSTEMS DEPT DAHLGREN VA 22448-5000
5	COMMANDER US ARMY ARDEC ATTN AMSTA AR CCH B E FENNELL A GONARTY T LOUZEIRO D KITCHEN P VALENTI PICATINNY ARSENAL NJ 07806-5000
1	COMMANDER US ARMY ARDEC ATTN AMSTA AR AET A B WONG PICATINNY ARSENAL NJ 07806-5000

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
38	DIR USARL ATTN AMSRL WT I MAY AMSRL WT P A HORST AMSRL WT PA T MINOR AMSRL WT PB E SCHMIDT A MIKHAIL J SAHU K HEAVY B GUIDOS P WEINACHT H EDGE (3 CPS) E FERRY V OSKAY J GARNER P PLOSTINS AMSRL WT PD B BURNS AMSRL WT W C MURPHY AMSRL WT WB W D'AMICO T BROWN B DAVIS E FERGUSON M HOLLIS AMSRL WT WC T VONG AMSRL SC W MERMAGEN D HISLEY AMSRL SC C W STUREK AMSRL SC AD C NIETUBICZ J GROSH P DYKSTRA D THOMPSON J HARE E MARK R ANGELINI K BURKE AMSRL SS IB J CLARKE (3 CPS)

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1093 (Edge) Date of Report June 1996

2. Date Report Received ____________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ____________________

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ____________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ____________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ____________________

CURRENT ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
U.S. ARMY RESEARCH LABORATORY
ATTN: AMSRL-WT-PB
ABERDEEN PROVING GROUND, MD 21005-5066